A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots

Yi Guo and Lynne E. Parker
Oak Ridge National Laboratory

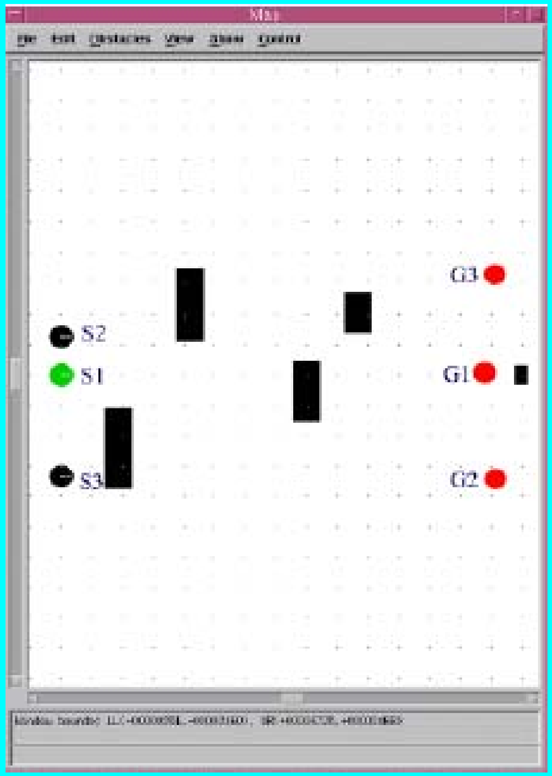
Proceedings 2002 IEEE International Conference on Robotics and Automation, May 2002, Washington, DC

Presentation by Ben Birch, March 6, 2003

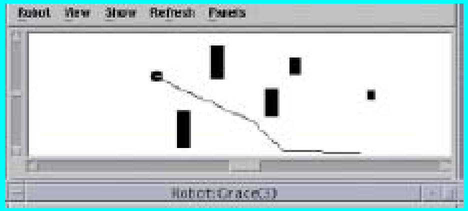
<u>The Problem</u>

Plan the motions of N robots as they move from their start to their goal positions without collisions with static obstacles and each other, while minimizing the following global performance index:

$$\Gamma \;=\; \gamma_1 \max(T_1, T_2, \ldots, T_N) + \gamma_2 \sum_{i=1}^{N} I_i$$



Motion planning consists of developing, for each robot, a geometric path and a velocity profile.

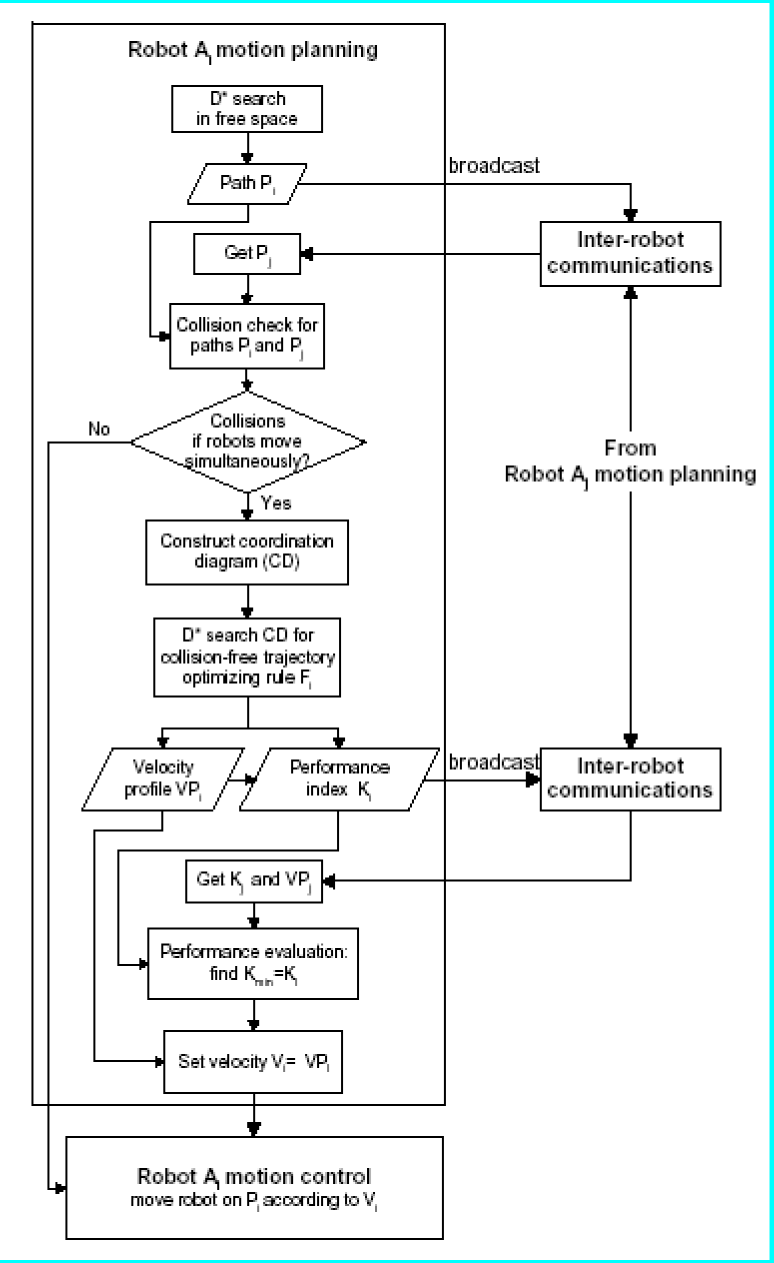


Assumptions

1. Each robot has an assigned goal and each robot knows its start and goal positions.
2. Robots operate in either indoor or outdoor environments and have a pre-defined map.
3. Robots' onboard sensors detect the discrepancy between the pre-defined map and the environment, and revise the map online.
4. Robots have perfect communication.
5. A robot's motion control layer tracks pre-assigned trajectories within a small margin of error.
6. Robots move at constant fixed speeds.
7. Robots can switch instantaneously between a fixed speed and halting.

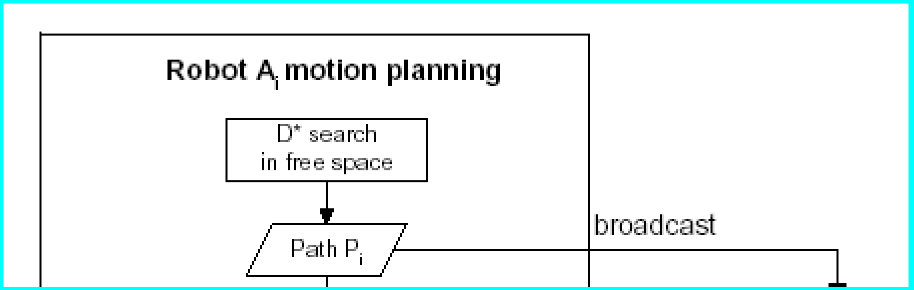
The Solution!!!

The solution involves decomposing the motion planning problem into two smaller subproblems: path planning and velocity planning.



D* Search in Free Space

The D* search algorithm is a dynamic version of A*.



The following cost function is used for D* path planning:

$$f_{pp} \quad = \quad \rho + \alpha_1 d + \alpha_2 s + \alpha_3 t$$

where:

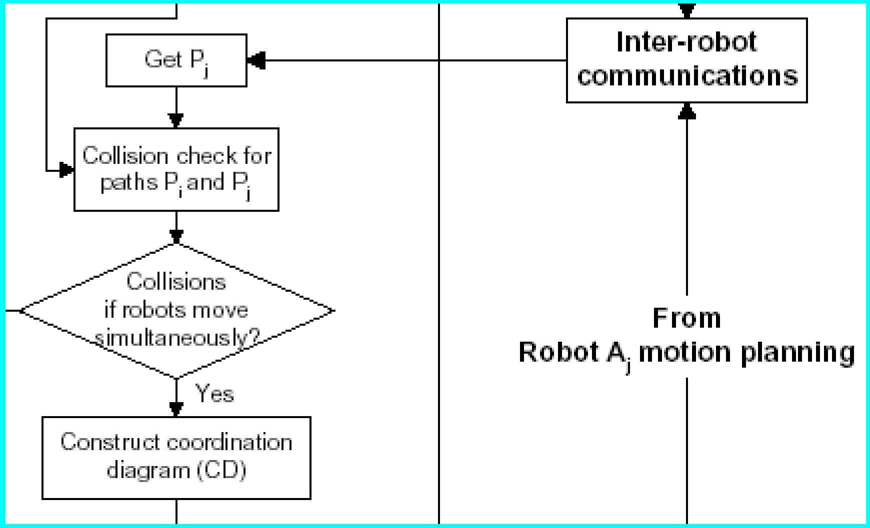
rho is a large value if obstacles are present, and 0 otherwise
d is the geometric distance
s is the slope of the terrain
t is the penalty for turning
alphas are positive weighting factors

Collision Check and Coordination Diagram

The collision check procedure returns all *collision regions* enlarged by the radius of the robot plus a margin of safety.

Each path is mapped to a one-dimensional trajectory based on path length.

All these mappings are combined into an N-dimensional coordination space represented by a *coordination diagram*.

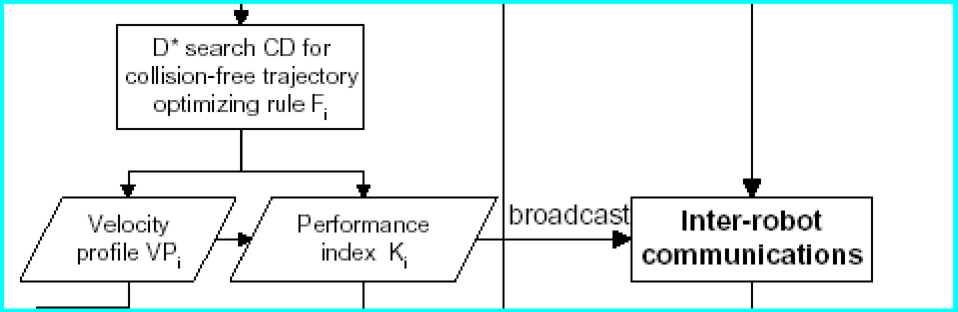


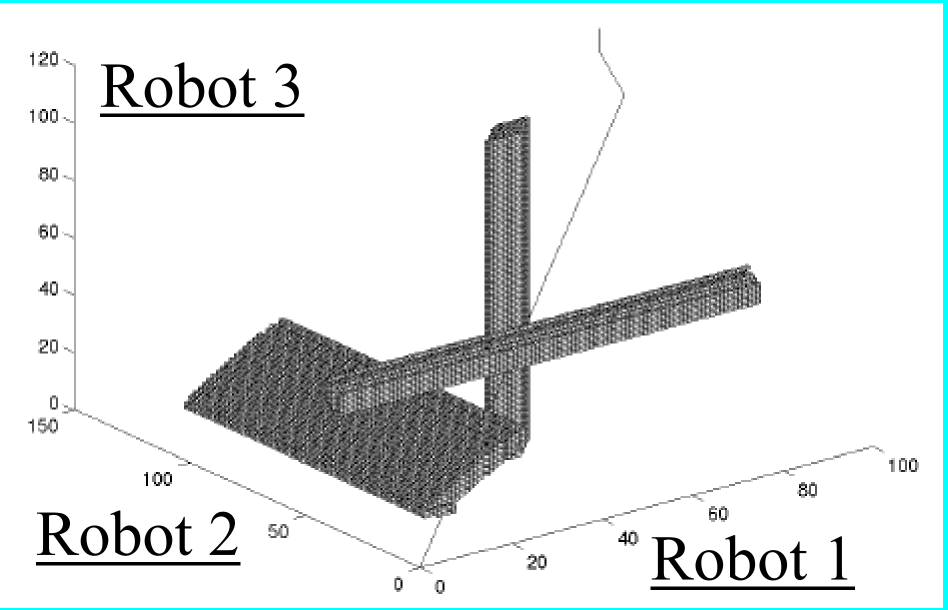
Search in Coordination Diagram

First the collision regions are mapped as static obstacles.

At each grid point, 2^N – 1 action combinations are considered.

The following cost function is used for D* search in the coordination diagram:





$$f_{vp} = \rho + \beta_1 d + \beta_2 t_{idle} + \beta_3 p$$

Where:

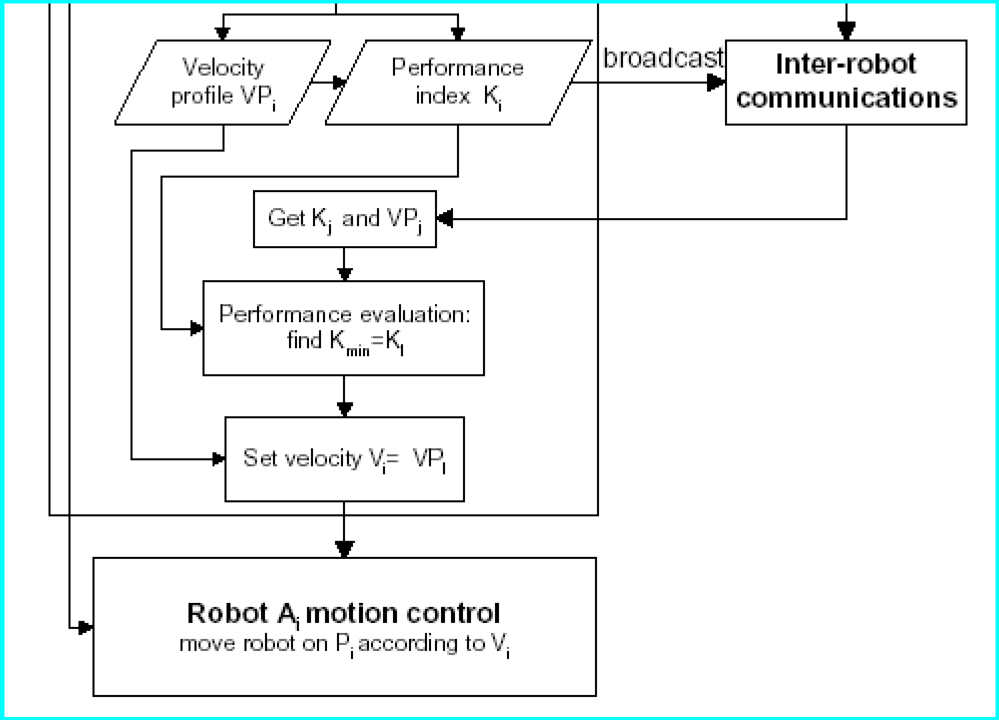
rho is a large value if obstacles are present, and 0 otherwise
d is the N-dimensional Euclidian distance
t is the total idle time for all robots
p is the penalty if this robot has to yield to other robots
beta are positive weighting factors

Selecting an Optimal Solution

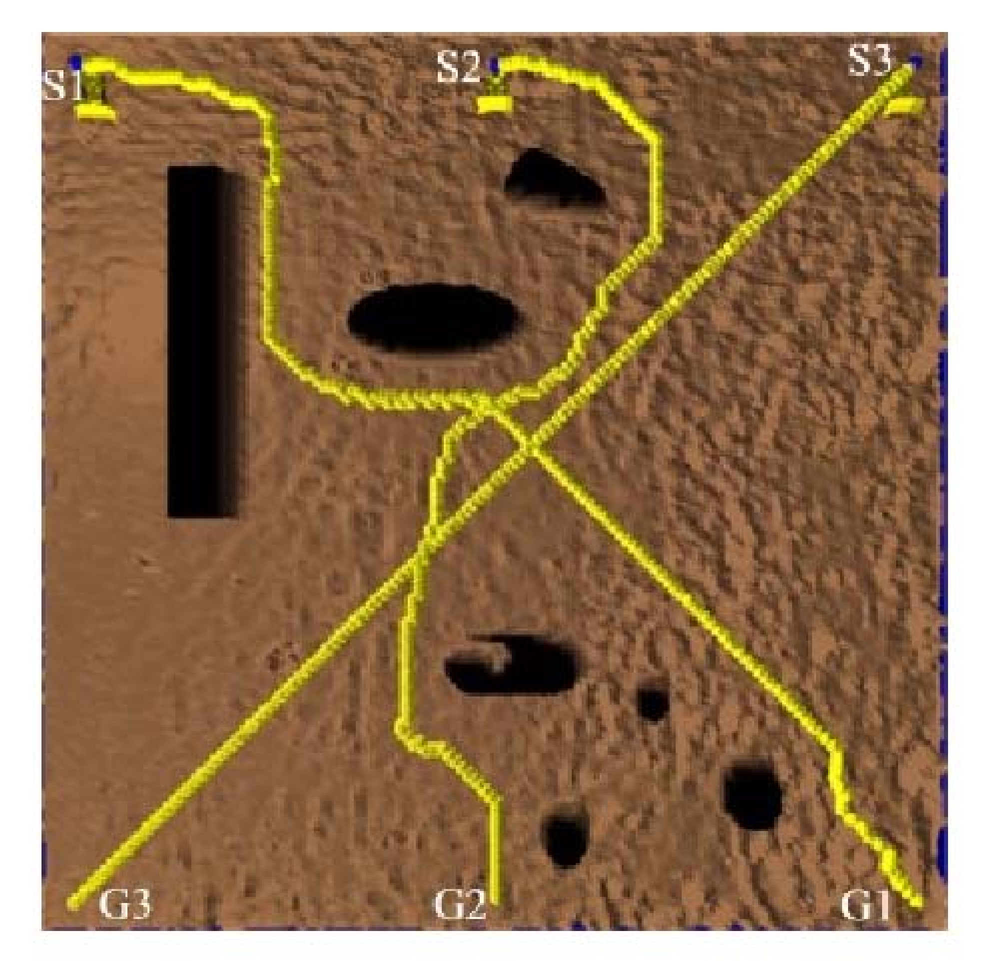
Each robot broadcasts the velocity profile and performance index that it has obtained.

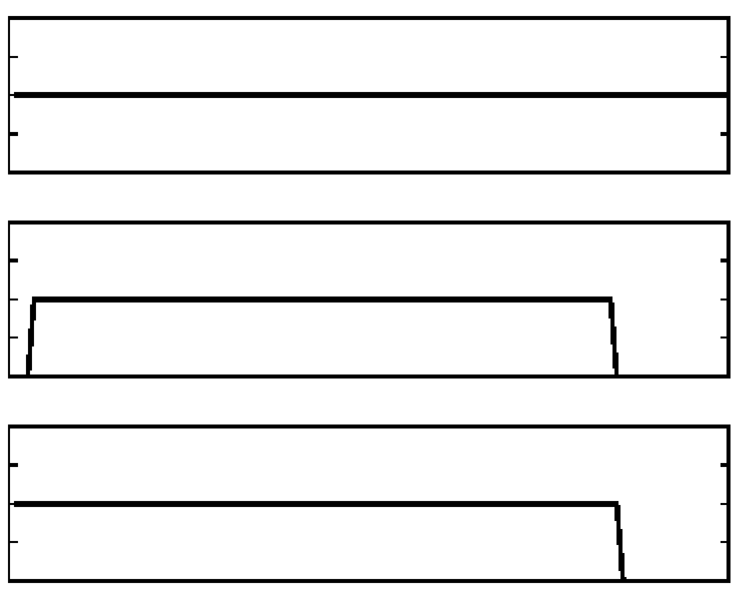
An evaluation is done to find the lowest performance index. Each robot is assigned the velocity profile associated with this index.

The robots then execute their paths according to their assigned velocity profile.

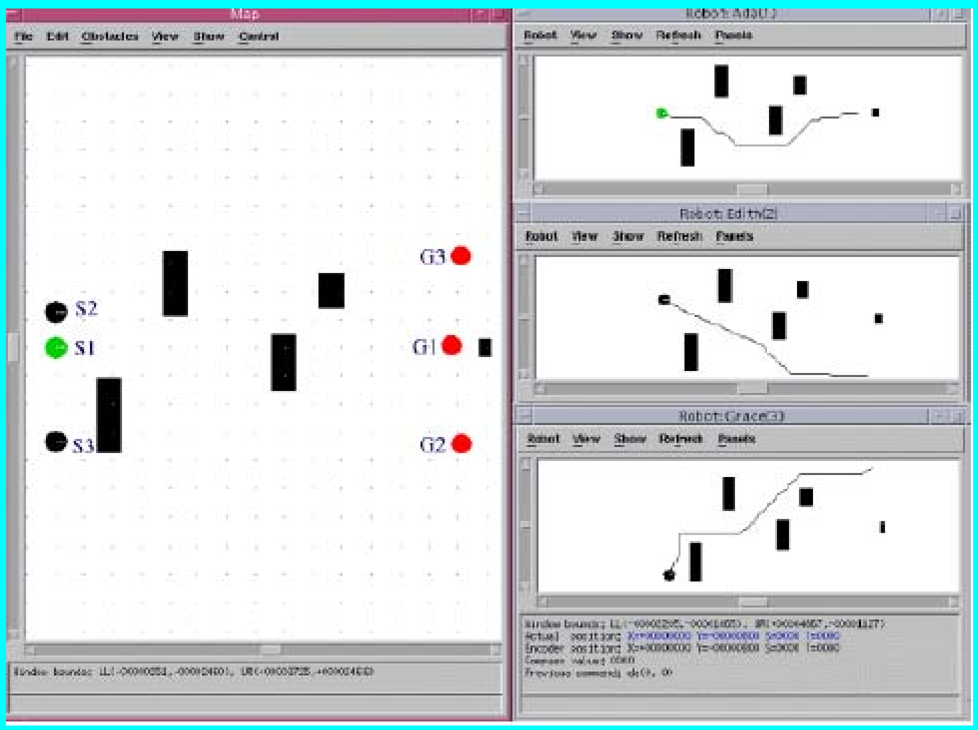


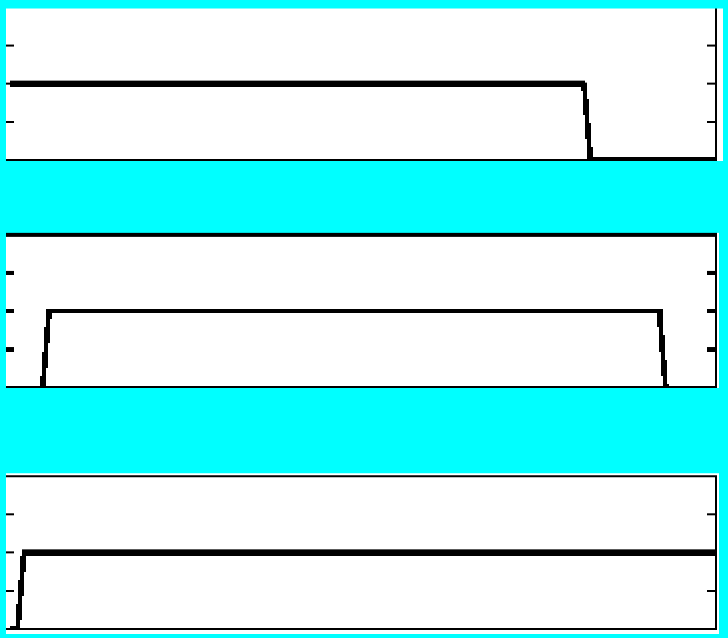
A 3D Simulation in an Outdoor Environment



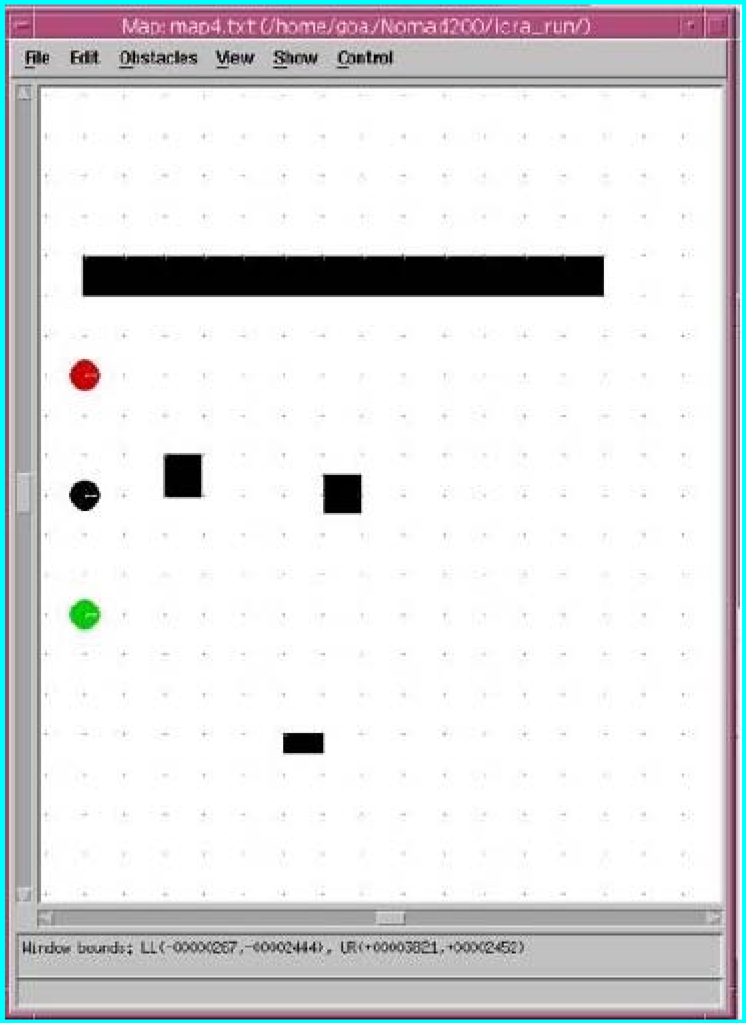


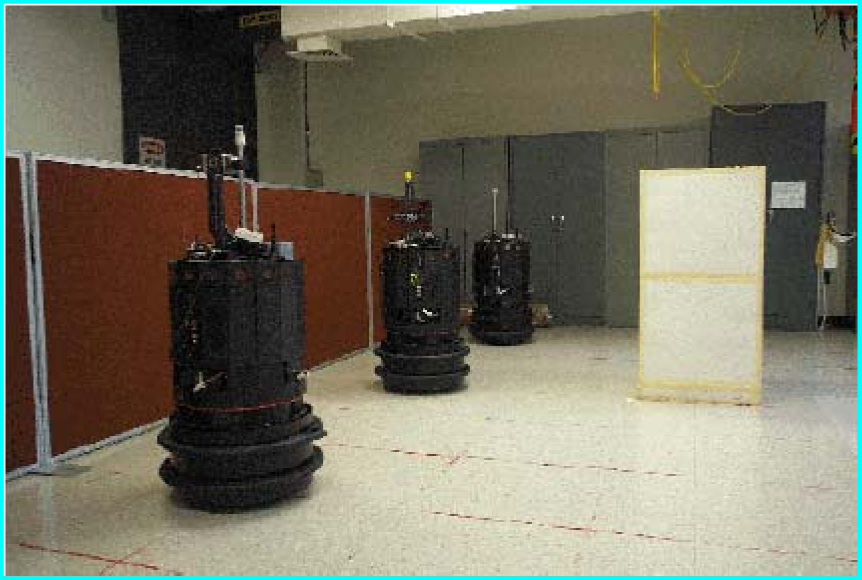
A 2D Simulation in an Indoor Environment





A Physical Robot Experiment





Contributions

Distributed AND Optimal!!!

Previous work achieved optimal solutions through centralized and exhaustive computing, or achieved distributed implementation without considering any optimization issues.

Handles terrain preferences and real time replanning.

These issues were seldom addressed in the previous literature.

Limitations

Computationally expensive for large numbers of robots ($2^N - 1$).

Untested on physical robots in outdoor environments, and for $N > 3$.

?

Questions

?